CLAIMS AS THEY STAND

1-36 (Canceled)

37. (Previously Presented) A method implemented by a server, comprising:

receiving a request from a first client to browse contents of a first file system on a first data server, wherein the first data server implements the first file system for managing file access and storage, and wherein the first client is unaware that the first data server implements the first file system;

selecting a first protocol interpreter from a plurality of different protocol interpreters, wherein the first protocol interpreter implements a first file access protocol which enables interaction with the first file system;

invoking the first protocol interpreter to interact with the first file system of the first data sever to obtain therefrom a first list of contents, wherein the first list of contents sets forth a hierarchical listing of at least a portion of the contents of the first file system on the first data server, the first list of contents comprising one or more directories and zero or more files; and

sending at least a portion of the first list of contents to the first client.

38. (Previously Presented) The method of claim 37, wherein the first client executes a web browser and submits the request using the web browser.

39. (Previously Presented) The method of claim 37, wherein the first client does not implement the first file access protocol such that the first client is incapable of interacting directly with the first file system.

40. (Previously Presented) The method of claim 37, further comprising:

receiving a request from the first client to browse contents of a second file system on a second data server, wherein the second data server implements the second file system, different from the first file system, for managing file access and storage, and wherein the first client is unaware that the second data server implements the second file system;

selecting a second protocol interpreter from the plurality of different protocol interpreters, wherein the second protocol interpreter implements a second file access protocol which enables interaction with the second file system;

invoking the second protocol interpreter to interact with the second file system of the second data sever to obtain therefrom a second list of contents, wherein the second list of contents sets forth a hierarchical listing of at least a portion of the contents of the second file system on the second data server, the second list of contents comprising one or more directories and zero or more files; and

sending at least a portion of the second list of contents to the first client.

41. (Previously Presented) The method of claim 40, wherein the first client does not implement the second file access protocol such that the first client is incapable of interacting directly with the second file system.

42. (Previously Presented) The method of claim 37, further comprising:

receiving a request from a second client to browse contents of a second file system on a second data server, wherein the second data server implements the second file system, different from the first file system, for managing file access and storage, and

wherein the second client is unaware that the second data server implements the second file system;

selecting a second protocol interpreter from the plurality of different protocol interpreters, wherein the second protocol interpreter implements a second file access protocol which enables interaction with the second file system;

invoking the second protocol interpreter to interact with the second file system of the second data sever to obtain therefrom a second list of contents, wherein the second list of contents sets forth a hierarchical listing of at least a portion of the contents of the second file system on the second data server, the second list of contents comprising one or more directories and zero or more files; and

sending at least a portion of the second list of contents to the second client.

43. (Previously Presented) The method of claim 42, wherein the second client does not implement the second file access protocol such that the second client is incapable of interacting directly with the second file system.

44. (Previously Presented) The method of claim 37, further comprising:

receiving a request from the first client to further explore a particular directory on the first data server;

invoking the first protocol interpreter to interact with the first file system of the first data sever to obtain therefrom a second list of contents, wherein the second list of contents comprises zero or more directories and one or more files stored within the particular directory; and

sending at least a portion of the second list of contents to the first client.

45. (Previously Presented) The method of claim 37, further comprising:

receiving a request from the first client to access a particular file stored on the first data server;

invoking the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom;

determining a file type for the particular file;

generating a set of encoding information based upon the file type of the particular file, wherein the set of encoding information comprises information for causing the first client to execute a particular type of application to process the particular file; and

sending the particular file and the set of encoding information to the first client.

46. (Previously Presented) The method of claim 45, wherein sending the particular file and the set of encoding information comprises:

sending the particular file and the set of encoding information as an electronic mail file to the first client.

47. (Previously Presented) The method of claim 46, wherein the electronic mail file comprises Multipurpose Internet Mail Extension (MIME) information.

48. (Previously Presented) The method of claim 45, wherein determining a file type for the particular file comprises:

determining a file extension for the particular file; and

processing the file extension to determine a file type for the particular file.

49. (Previously Presented) The method of claim 48, wherein processing comprises:

determining a Multipurpose Internet Mail Extension (MIME) type for the file extension.

50. (Previously Presented) The method of claim 49, wherein the set of encoding information comprises the MIME type.

51. (Previously Presented) The method of claim 37, further comprising:

receiving a request from the first client to compress a particular file stored on the first data server;

invoking the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom;

compressing the particular file to derive a compressed version; and

invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to store the compressed version onto the first data server.

52. (Previously Presented) The method of claim 37, further comprising:

receiving a request from the first client to send a particular file stored on the first data server to a recipient;

invoking the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom; and

sending the particular file to the recipient without first downloading the particular file to the first client.

53. (Previously Presented) The method of claim 37, further comprising:

receiving a request from the first client to create a new directory on the first data server; and

invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to create the new directory on the first data server.

54. (Previously Presented) The method of claim 37, further comprising:

receiving a request from the first client to store a new file onto the first data server;

receiving the new file from the first client; and

invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to store the new file onto the first data server.

55. (Previously Presented) The method of claim 37, further comprising:

receiving a search request from the first client comprising a set of search criteria;

processing the set of search criteria to derive one or more search commands;

invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to implement the one or more search commands; and

receiving one or more sets of search results from the first file system.

56. (Previously Presented) The method of claim 55, further comprising:

processing the one or more sets of search results to derive a processed set of search results; and

sending the processed set of search results to the first client.

57. (Previously Presented) The method of claim 55, wherein the set of search criteria comprises a Boolean operation and/or a wild card.

58. (Previously Presented) An apparatus, comprising:

a mechanism for receiving a request from a first client to browse contents of a first file system on a first data server, wherein the first data server implements the first file system for managing file access and storage, and wherein the first client is unaware that the first data server implements the first file system;

a mechanism for selecting a first protocol interpreter from a plurality of different protocol interpreters, wherein the first protocol interpreter implements a first file access protocol which enables interaction with the first file system;

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data sever to obtain therefrom a first list of contents, wherein the first list of contents sets forth a hierarchical listing of at least a portion of the contents of the first file system on the first data server, the first list of contents comprising one or more directories and zero or more files; and

a mechanism for sending at least a portion of the first list of contents to the first client.

59. (Previously Presented) The apparatus of claim 58, wherein the first client executes a web browser and submits the request using the web browser.

60. (Previously Presented) The apparatus of claim 58, wherein the first client does not implement the first file access protocol such that the first client is incapable of interacting directly with the first file system.

61. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from the first client to browse contents of a second file system on a second data server, wherein the second data server implements the second file system, different from the first file system, for managing file access and storage, and wherein the first client is unaware that the second data server implements the second file system;

a mechanism for selecting a second protocol interpreter from the plurality of different protocol interpreters, wherein the second protocol interpreter implements a second file access protocol which enables interaction with the second file system;

a mechanism for invoking the second protocol interpreter to interact with the second file system of the second data sever to obtain therefrom a second list of contents, wherein the second list of contents sets forth a hierarchical listing of at least a portion of the contents of the second file system on the second data server, the second list of contents comprising one or more directories and zero or more files; and

a mechanism for sending at least a portion of the second list of contents to the first client.

62. (Previously Presented) The apparatus of claim 61, wherein the first client does not implement the second file access protocol such that the first client is incapable of interacting directly with the second file system.

63. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from a second client to browse contents of a second file system on a second data server, wherein the second data server implements the second file system, different from the first file system, for managing file access and storage, and wherein the second client is unaware that the second data server implements the second file system;

a mechanism for selecting a second protocol interpreter from the plurality of different protocol interpreters, wherein the second protocol interpreter implements a second file access protocol which enables interaction with the second file system;

a mechanism for invoking the second protocol interpreter to interact with the second file system of the second data sever to obtain therefrom a second list of contents, wherein the second list of contents sets forth a hierarchical listing of at least a portion of the contents of the second file system on the second data server, the second list of contents comprising one or more directories and zero or more files; and

a mechanism for sending at least a portion of the second list of contents to the second client.

64. (Previously Presented) The apparatus of claim 63, wherein the second client does not implement the second file access protocol such that the second client is incapable of interacting directly with the second file system.

65. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from the first client to further explore a particular directory on the first data server;

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data sever to obtain therefrom a second list of contents, wherein the second list of contents comprises zero or more directories and one or more files stored within the particular directory; and

a mechanism for sending at least a portion of the second list of contents to the first client.

66. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from the first client to access a particular file stored on the first data server;

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom;

a mechanism for determining a file type for the particular file;

a mechanism for generating a set of encoding information based upon the file type of the particular file, wherein the set of encoding information comprises information for causing the first client to execute a particular type of application to process the particular file; and

a mechanism for sending the particular file and the set of encoding information to the first client.

67. (Previously Presented) The apparatus of claim 66, wherein the mechanism for sending the particular file and the set of encoding information comprises:

a mechanism for sending the particular file and the set of encoding information as an electronic mail file to the first client.

68. (Previously Presented) The apparatus of claim 67, wherein the electronic mail file comprises Multipurpose Internet Mail Extension (MIME) information.

69. (Previously Presented) The apparatus of claim 66, wherein the mechanism for determining a file type for the particular file comprises:

a mechanism for determining a file extension for the particular file; and

a mechanism for processing the file extension to determine a file type for the particular file.

70. (Previously Presented) The apparatus of claim 69, wherein the mechanism for processing comprises:

a mechanism for determining a Multipurpose Internet Mail Extension (MIME) type for the file extension.

71. (Previously Presented) The apparatus of claim 70, wherein the set of encoding information comprises the MIME type.

72. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from the first client to compress a particular file stored on the first data server;

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom;

a mechanism for compressing the particular file to derive a compressed version; and

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to store the compressed version onto the first data server.

73. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from the first client to send a particular file stored on the first data server to a recipient;

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom; and

a mechanism for sending the particular file to the recipient without first downloading the particular file to the first client.

74. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from the first client to create a new directory on the first data server; and

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to create the new directory on the first data server.

75. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a request from the first client to store a new file onto the first data server;

a mechanism for receiving the new file from the first client; and

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to store the new file onto the first data server.

76. (Previously Presented) The apparatus of claim 58, further comprising:

a mechanism for receiving a search request from the first client comprising a set of search criteria;

a mechanism for processing the set of search criteria to derive one or more search commands;

a mechanism for invoking the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to implement the one or more search commands; and

a mechanism for receiving one or more sets of search results from the first file system.

77. (Previously Presented) The apparatus of claim 76, further comprising:

a mechanism for processing the one or more sets of search results to derive a processed set of search results; and

a mechanism for sending the processed set of search results to the first client.

78. (Previously Presented) The apparatus of claim 76, wherein the set of search criteria comprises a Boolean operation and/or a wild card.

79. (Previously Presented) A computer readable medium, comprising:

instructions for causing one or more processors to receive a request from a first client to browse contents of a first file system on a first data server, wherein the first data server implements the first file system for managing file access and storage, and wherein the first client is unaware that the first data server implements the first file system;

instructions for causing one or more processors to select a first protocol interpreter from a plurality of different protocol interpreters, wherein the first protocol interpreter implements a first file access protocol which enables interaction with the first file system;

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data sever to obtain therefrom a first list of contents, wherein the first list of contents sets forth a hierarchical listing of at least a portion of the contents of the first file system on the first data server, the first list of contents comprising one or more directories and zero or more files; and

instructions for causing one or more processors to send at least a portion of the first list of contents to the first client.

80. (Previously Presented) The computer readable medium of claim 79, wherein the first client executes a web browser and submits the request using the web browser.

81. (Previously Presented) The computer readable medium of claim 79, wherein the first client does not implement the first file access protocol such that the first client is incapable of interacting directly with the first file system.

82. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from the first client to browse contents of a second file system on a second data server, wherein the second data server implements the second file system, different from the first file system, for managing file access and storage, and wherein the first client is unaware that the second data server implements the second file system;

instructions for causing one or more processors to select a second protocol interpreter from the plurality of different protocol interpreters, wherein the second protocol interpreter implements a second file access protocol which enables interaction with the second file system;

instructions for causing one or more processors to invoke the second protocol interpreter to interact with the second file system of the second data sever to obtain therefrom a second list of contents, wherein the second list of contents sets forth a hierarchical listing of at least a portion of the contents of the second file system on the

second data server, the second list of contents comprising one or more directories and zero or more files; and

instructions for causing one or more processors to send at least a portion of the second list of contents to the first client.

83. (Previously Presented) The computer readable medium of claim 82, wherein the first client does not implement the second file access protocol such that the first client is incapable of interacting directly with the second file system.

84. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from a second client to browse contents of a second file system on a second data server, wherein the second data server implements the second file system, different from the first file system, for managing file access and storage, and wherein the second client is unaware that the second data server implements the second file system;

instructions for causing one or more processors to select a second protocol interpreter from the plurality of different protocol interpreters, wherein the second protocol interpreter implements a second file access protocol which enables interaction with the second file system;

instructions for causing one or more processors to invoke the second protocol interpreter to interact with the second file system of the second data sever to obtain therefrom a second list of contents, wherein the second list of contents sets forth a hierarchical listing of at least a portion of the contents of the second file system on the

second data server, the second list of contents comprising one or more directories and zero or more files; and

instructions for causing one or more processors to send at least a portion of the second list of contents to the second client.

85. (Previously Presented) The computer readable medium of claim 84, wherein the second client does not implement the second file access protocol such that the second client is incapable of interacting directly with the second file system.

86. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from the first client to further explore a particular directory on the first data server;

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data sever to obtain therefrom a second list of contents, wherein the second list of contents comprises zero or more directories and one or more files stored within the particular directory; and

instructions for causing one or more processors to send at least a portion of the second list of contents to the first client.

87. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from the first client to access a particular file stored on the first data server;

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom;

instructions for causing one or more processors to determine a file type for the particular file;

instructions for causing one or more processors to generate a set of encoding information based upon the file type of the particular file, wherein the set of encoding information comprises information for causing the first client to execute a particular type of application to process the particular file; and

instructions for causing one or more processors to send the particular file and the set of encoding information to the first client.

88. (Previously Presented) The computer readable medium of claim 87, wherein the instructions fro causing one or more processors to send the particular file and the set of encoding information comprises:

instructions for causing one or more processors to send the particular file and the set of encoding information as an electronic mail file to the first client.

89. (Previously Presented) The computer readable medium of claim 88, wherein the electronic mail file comprises Multipurpose Internet Mail Extension (MIME) information.

90. (Previously Presented) The computer readable medium of claim 87, wherein the instructions for causing one or more processors to determine a file type for the particular file comprises:

instructions for causing one or more processors to determine a file extension for the particular file; and

instructions for causing one or more processors to process the file extension to determine a file type for the particular file.

91. (Previously Presented) The computer readable medium of claim 90, wherein the instructions for causing one or more processors to process comprises:

instructions for causing one or more processors to determine a Multipurpose Internet Mail Extension (MIME) type for the file extension.

92. (Previously Presented) The computer readable medium of claim 91, wherein the set of encoding information comprises the MIME type.

93. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from the first client to compress a particular file stored on the first data server;

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom;

instructions for causing one or more processors to compress the particular file to derive a compressed version; and

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to store the compressed version onto the first data server.

94. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from the first client to send a particular file stored on the first data server to a recipient;

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data server to retrieve the particular file therefrom; and

instructions for causing one or more processors to send the particular file to the recipient without first downloading the particular file to the first client.

95. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from the first client to create a new directory on the first data server; and

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to create the new directory on the first data server.

96. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a request from the first client to store a new file onto the first data server;

instructions for causing one or more processors to receive the new file from the first client; and

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to store the new file onto the first data server.

97. (Previously Presented) The computer readable medium of claim 79, further comprising:

instructions for causing one or more processors to receive a search request from the first client comprising a set of search criteria;

instructions for causing one or more processors to process the set of search criteria to derive one or more search commands;

instructions for causing one or more processors to invoke the first protocol interpreter to interact with the first file system of the first data server to cause the first file system to implement the one or more search commands; and

instructions for causing one or more processors to receive one or more sets of search results from the first file system.

98. (Previously Presented) The computer readable medium of claim 97, further comprising:

instructions for causing one or more processors to process the one or more sets of search results to derive a processed set of search results; and

instructions for causing one or more processors to send the processed set of search results to the first client.

99. (Previously Presented) The computer readable medium of claim 97, wherein the set of search criteria comprises a Boolean operation and/or a wild card.